INSTITUT FÜR INFORMATIK

Aggressive Space Mapping for the Optimization of a Marine Ecosystem Model

Malte Prieß, Thomas Slawig

Bericht Nr. 1014
November 11, 2010

CHRISTIAN-ALBRECHTS-UNIVERSITÄT

ZU KIEL

Institut für Informatik der
Christian-Albrechts-Universität zu Kiel
Olshausenstr. 40
D – 24098 Kiel

Aggressive Space Mapping for the Optimization of a Marine Ecosystem Model

Malte Prieß, Thomas Slawig

Bericht Nr. 1014
November 11, 2010

e-mail: mpr@informatik.uni-kiel.de, ts@informatik.uni-kiel.de

Aggressive Space Mapping for the Optimization of a Marine Ecosystem Model

Malte Prieß*, Thomas Slawig†

Abstract

In this paper we apply the Aggressive Space Mapping (ASM) algorithm by Bandler et. al. (cf. [1]) to the parameter optimization of a one-dimensional marine ecosystem model of NPZD type. We show that this approach leads to a very satisfactory solution while yielding a significant reduction in the total optimization cost. The ecosystem model, developed by Oschlies and Garcon [8], simulates the distribution of nitrogen, phytoplankton, zooplankton and detritus in a water column and is driven by ocean circulation data. A key issue is to optimize model parameters in order to minimize the misfit between the model output and given observational data. In the ASM approach, reducing the overall optimization cost by avoiding expensive function and derivative evaluations is achieved by using a surrogate model that replaces the original one. Furthermore the ASM algorithm solves a nonlinear system of equations which is conditionally equivalent to use this surrogate in the optimization run. We use a coarser time discretization for obtaining a suitable low-fidelity model. This is then corrected to create a physically-based surrogate, where the correction is obtained through a parameter mapping which provides the minimizer of the distance between the fine and the coarse model output. We show that this surrogate provides a good approximation of the fine model. The applicability of the ASM technique to the problem at hand is verified by using synthetic target data. Results are compared to those of the direct fine model optimization. We show that a very reasonable fit of the target data can be obtained with an average reduction in the computational cost of about 65%.

Key words: Marine ecosystem models, surrogate-based optimization, low-fidelity model, Aggressive Space Mapping, coarse discretization, Globalized Quasi-Newton method.

1 Introduction

Understanding the oceanic CO_2 uptake is of central importance for projections of climate change and oceanic ecosystems. Simulating ocean circulation and biogeochemistry has become a key tool for understanding the ocean carbon cycle and its variability. The underlying models are governed by coupled systems of parabolic partial differential equations for ocean circulation (ocean models) and transport of biogeochemical tracers (marine ecosystem models). The coupling relations between the tracers are more or less empirical,

*mpr@informatik.uni-kiel.de

†ts@informatik.uni-kiel.de, both: Department of Computer Science, Algorithmic Optimal Control – CO_2 Uptake of the Ocean, Excellence Cluster The Future Ocean, Christian-Albrechts-Platz 4, 24118 Kiel, Germany.

i.e. it is not very clear how the coupling terms look like mathematically, and, moreover, how many tracers have to be taken into account. In ecosystem models many parameters are used which are chosen such that given measurement data are matched, and that the model output remains feasible (i.e. non-negative).

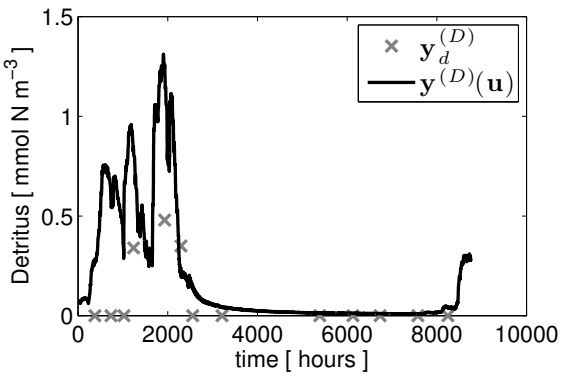

Figure 1: Model output $\mathbf{y}^{(D)}$ (detritus) and target data $\mathbf{y}_d^{(D)}$ for one year at depth $z \approx 25\ m$.

For this purpose the aim is to minimize a least-squares type cost functional, measuring this misfit (cf. Figure 1), and optionally constrained by inequalities for parameters and/or state variables, i.e. tracers. The optimization variables are unknown physical/biological parameters in the nonlinear coupling terms in the tracer transport equations. This optimization process requires a lot of – maybe expensive – function and optionally sensitivity or gradient evaluations. If the latter are computed by finite difference approximations, the critical quantity regarding the computational cost of the optimization is the one needed for one function evaluation, which is basically one model run. Hence a big issue in order to reduce the overall optimization cost is to decrease the effort for the function evaluations. This in particular becomes significant for computationally expensive three-dimensional coupled models.

The idea of surrogate optimization is to replace the original model in focus (also called *high-fidelity* or *fine model*) by a computationally cheaper so-called *surrogate*. To create this surrogate a *low-fidelity* or *coarse model*, which is usually less accurate, is introduced and iteratively corrected by suitable methods. For this correction or alignment, only a few evaluations of the high-fidelity model and possibly also its derivatives are necessary. Apart from this alignment, the whole optimization process is performed in the surrogate's model space which could dramatically reduce the overall cost.

In this paper we particularly analyze the application of the so-called Aggressive Space Mapping (ASM) technique (firstly developed by Bandler et. al [1, 2]) to the optimization of a one-dimensional ecosystem model which is introduced in Section 2. In Section 3 we describe the corresponding optimization problem. The basic idea of surrogate-based optimization is recalled in Section 4. We use a coarsening in the temporal mesh to create the basis of our surrogate, the low-fidelity model, which we briefly present in Section 5. This low-fidelity model is then corrected by a so-called parameter mapping (cf. [2]) to obtain our surrogate model which we describe in Section 6. The basic idea of the ASM algorithm and the globalized Quasi-Newton procedure [6] we use to compute the ASM solution are then presented in the Section 7. To verify the approach we test the algorithm by using synthetic target data and compare the results with those obtained from direct fine model optimization. Corresponding results are given in Section 8. We draw some conclusions in Section 9.

2 Model Description

A one-dimensional marine ecosystem model that simulates the interaction of dissolved inorganic nitrogen, phytoplankton, zooplankton and detritus (thus also called $NPZD$

model) was developed by Oschlies and Garcon [8], with the aim of simultaneously reproducing observations at three North Atlantic locations by the optimization of free parameters within credible limits. The model uses the ocean circulation and temperature field in an off-line modus, i.e. no feedback on them is modeled. The model simulates one water column at a given horizontal position which is motivated by the fact that there have been special time series studies at fixed locations. Marine ecosystem models are coupled PDE systems consisting of time-dependent advection-diffusion-reaction equations with nonlinear coupling terms. The velocity and temperature and sometimes also salinity data are either computed simultaneously or in advance by an ocean model. Clearly, the second variant, that is used in this paper, is computationally cheaper but neglects the feedback effects from the biogeochemistry to the ocean circulation and temperature distribution etc.

In the model, the concentrations (in mmol N m^{-3}) of dissolved inorganic nitrogen (DIN), phytoplankton (PHY), zooplankton (ZOO), and detritus (DET) are summarized in the vector $\mathbf{y} := (y^{(l)})_{l=N,P,Z,D}$ and described by the following coupled PDE system

$$\left.\begin{array}{rcll} \dfrac{\partial y^{(l)}}{\partial t} & = & \dfrac{\partial}{\partial z}\left(K_\rho \dfrac{\partial y^{(l)}}{\partial z}\right) & + \quad Q^{(l)}(\mathbf{y}, t, T, u_2, \ldots, u_n), \qquad l = N, P, Z \\ \dfrac{\partial y^{(D)}}{\partial t} & = & \dfrac{\partial}{\partial z}\left(K_\rho \dfrac{\partial y^{(D)}}{\partial z}\right) & + \quad Q^{(D)}(\mathbf{y}, t, T, u_2, \ldots, u_n) - \dfrac{\partial y^{(D)}}{\partial z} u_1, \quad l = D \end{array}\right\} \quad (1)$$

$$\text{in } (0, H) \times (0, t_e)$$

with additional appropriate initial values. Here z denotes the vertical spatial coordinate, H the depth of the water column, and t_e the total integration time. The $Q^{(l)}$ are the biogeochemical coupling (or "source-minus-sink") terms for the four tracers, and $\mathbf{u} = (u_1, \ldots, u_n)$ is the vector of unknown physical and biological parameters. The sinking term is only apparent in the equation for detritus. In the one-dimensional model no advection term is used, since a reduction to vertical advection only would make no sense. Thus the circulation data (taken from an ocean model) are the turbulent mixing coefficient $K_\rho = K_\rho(z, t)$ and the temperature $T = T(z, t)$.

3 Optimization Problem

The aim is to minimize a least-squares type cost function measuring the misfit between the model output $\mathbf{y} = \mathbf{y}(\mathbf{u})$ and given observational data $\mathbf{y}_d$ (cf. Figure (1)). The optimization variables are summarized in the vector $\mathbf{u}$ of unknown parameters. Thus the problem can be written as

$$\min_{\mathbf{u} \in U} J(\mathbf{y}(\mathbf{u}), \mathbf{u}) := || \mathbf{y}(\mathbf{u}) - \mathbf{y}_d ||_Y^2 + \alpha || \mathbf{u} ||_U^2$$

$$U := \{\mathbf{u} \in \mathbb{R}^n : \mathbf{b}_l \leq \mathbf{u} \leq \mathbf{b}_u\} \quad , \quad J : Y \times U \to \mathbb{R}. \quad (2)$$

Note that the optimization variables are real numbers. Since most of the parameters are growth or dying rates, component-wise lower and upper bounds described by the vectors $\mathbf{b}_l, \mathbf{b}_u \in \mathbb{R}^n$ are imposed on them.

The above formulation of the optimization problem is valid if the state or vector of tracers $\mathbf{y}$ is regarded

- either in a continuous setting as an element of an appropriately chosen function space Y,
- or after discretization (cf. Section 5) as a discrete vector $\mathbf{y} \in \mathbb{R}^{4KM} = Y$ with K and M denoting the total number of spatial and temporal grid points, respectively.

We will consider the latter formulation from now on.

Constraints on the state variable $\mathbf{y}$ are not treated explicitly in our formulation (2). However, by using appropriate parameter bounds $\mathbf{b}_l$ and $\mathbf{b}_u$, the desired non-negativity of the state/tracer vector can be ensured in the model. This was already observed and used in [9].

4 Surrogate-Based Optimization

For many nonlinear optimization problems high computational cost of accurate simulations and derivatives or even the lack of sensitivity information are major drawbacks. The need for a decrease in the computational cost is especially important in the case of the optimization of complex three-dimensional models.

In surrogate-based optimization, the original *high-fidelity model* output $\mathbf{y}$ is replaced by a surrogate which is based on the output $\hat{\mathbf{y}}$ of a *coarse model* and subsequently aligned or updated. In the kth step of an optimization algorithm, arrived at optimization variable iterate $\mathbf{u}_k \in U$, we thus perform a step generally written as

$$\mathbf{y}(\mathbf{u}_k)\ ,\ \hat{\mathbf{y}}(\mathbf{u}_k) \quad \mapsto \quad \mathbf{s}_k. \tag{3}$$

The surrogate $\mathbf{s}_k$ should satisfy so-called 0-order and ideally also 1st-order consistency with the high-fidelity model in the current iterate $\mathbf{u}_k$, i.e.

$$\mathbf{s}_k(\mathbf{u}_k) \approx \mathbf{y}(\mathbf{u}_k) \quad , \quad \mathbf{s}_k'(\mathbf{u}_k) \approx \mathbf{y}'(\mathbf{u}_k), \tag{4}$$

and maybe also in a neighborhood.

Key points for a well performing surrogate algorithm are a cheap function and sensitivity evaluation of its basis, the low-fidelity model, a low cost for the alignment of this low-fidelity model, and a low number of necessary iterations in the surrogate-based optimization process, since this results in only few evaluations of the high-fidelity model.

5 The Low-Fidelity Model

Surrogates can be either based upon a functional or a physical low-fidelity model. Those obtained from a functional low-fidelity model are constructed without any particular knowledge of the system on the basis of sampled data of the fine-model only and will not be addressed further in this paper. In contrast, surrogates based upon a physical low-fidelity model (also known as *physical surrogates*) inherit more characteristics of the fine model in focus. Possible ways to create such a physical low-fidelity model are by a coarser discretization, by using simplified physics or different ways of describing the same physical phenomenon or even by using analytical formulas if available. In this paper we use a low-fidelity model which has a coarser time discretization. For this purpose, we now present the original discretization scheme of the $NPZD$ model.

Discretization Scheme

The original or *high-fidelity model* described by (1) is solved using an operator splitting method: Given a time-step τ, the discretized scheme reads

$$\underbrace{\left[I - \tau A_j^{\text{diff}}\right]}_{:=B_j^{\text{diff}}} \mathbf{y}_{j+1} = \underbrace{\left[I + \tau \cdot A^{\text{sink}}\right]}_{:=B^{\text{sink}}} \circ B_j^Q \circ B_j^Q \circ B_j^Q \circ B_j^Q(\mathbf{y}_j), \quad j = 1, \ldots, M. \quad (5)$$

Here we denote by $\mathbf{y}_j \approx (\, y(z_i, t_j)\,)_{i=1,\ldots,K}$ the discrete solution in time step j, where K is the number of discrete spatial points.

At first, the nonlinear coupling operators Q_j are computed at every spatial grid point and integrated by four explicit Euler steps, each of which is described by the operator

$$B_j^Q(\mathbf{y}_j) \;:=\; \left[I + \frac{\tau}{4} Q_j(\mathbf{y}_j)\right]. \quad (6)$$

Here we omitted the additional arguments of the Q_j for simplicity.

Then, an explicit Euler step with full step-size τ is performed for the sinking term which is spatially discretized by an upstream scheme. This step is summarized in the matrix B^{sink}. Since the sinking velocity is temporal constant, this matrix does not depend on the time step j.

Finally, an implicit Euler step for the diffusion operator, discretized with second order central differences, is applied. Due to $K_\rho = K_\rho(z,t)$ the resulting matrix B_j^{diff} depends on j and is non-symmetric, cf. [5, Section 5]. It is tridiagonal, and the system is solved directly. Note that $A_j^{\text{diff}}, A^{\text{sink}}$ are 4×4 block-diagonal matrices.

Coarser Time Discretization

The low-fidelity model is obtained by using a coarser time discretization with

$$\hat{\tau} = \beta\tau$$

with a *coarsening factor* β in the range $[10, 60]$, while keeping the spatial discretization fixed. The state variable for this coarser discretized model will be denoted by $\hat{\mathbf{y}}$, the corresponding number of discrete time steps by $\hat{M} = M/\beta$. The parameters $\mathbf{u}$ for this coarse model are the same as for the fine model. Figure 2 shows the fine and coarse model output $\mathbf{y}^{(D)}, \hat{\mathbf{y}}^{(D)}$ for the state detritus, for different values of β and at the same randomly chosen parameter vector $\mathbf{u}$.

It is important to keep in mind that choosing β too big could lead to a numerically unstable scheme. The condition on stability is determined by the ratio h/u_1 where h denotes the size of the discrete spatial step. All computations in this paper were performed with parameters that guarantee stability.

6 The Surrogate Model

The surrogate model we use is obtained by a space mapping approach introduced by Bandler et al. [1]. The physical low-fidelity model with output $\hat{\mathbf{y}}$ (cf. Section 5) is corrected in the kth optimization step by a so-called parameter mapping $\mathbf{p}_k$ to obtain a surrogate $\mathbf{s}_k$ for the fine model, in detail

$$\mathbf{s}_k(\mathbf{u}) := \hat{\mathbf{y}}\left[\mathbf{p}_k(\mathbf{u})\right] \quad , \quad \mathbf{p}_k(\mathbf{u}) = \mathbf{p}(\mathbf{u}_k) + \mathbf{p}'(\mathbf{u}_k)\,(\mathbf{u} - \mathbf{u}_k)$$

$$\hat{\mathbf{u}}_k = \mathbf{p}(\mathbf{u}_k) := \text{argmin}_{\mathbf{u}\in U} \;\; ||\,\hat{\mathbf{y}}(\mathbf{u}) - \mathbf{y}(\mathbf{u}_k)\,||_Y^2\,. \quad (7)$$

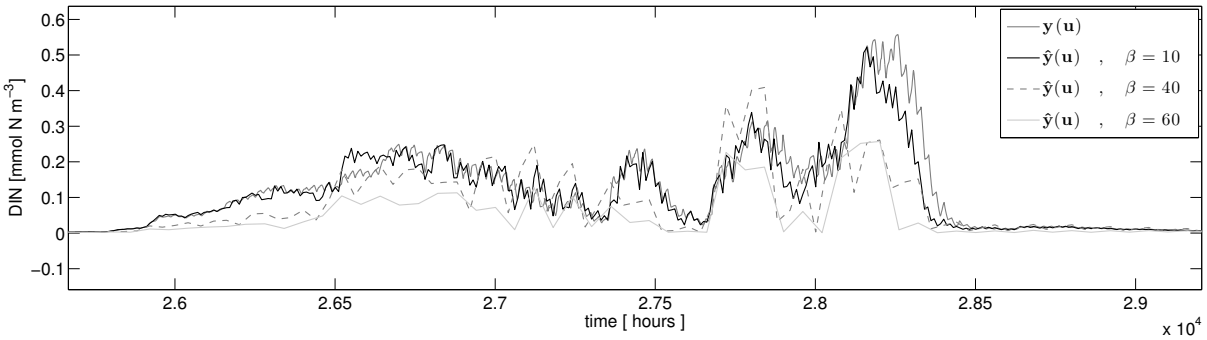

Figure 2: High- and low-fidelity model output $\mathbf{y}^{(N)}, \hat{\mathbf{y}}^{(N)}$, respectively, for the state dissolved inorganic nitrogen DIN at depth $z \approx 2.68m$ for different values of the coarsening factor β and the same randomly chosen parameter vector $\mathbf{u}$. For simplicity we skip super- and subscripts in the legends of all figures.

The usually non-linear mapping $\mathbf{p}$ is aligning the high- and low-fidelity model and is approximated in the point $\mathbf{u}_k$ using a first-order Taylor expansion.

0-order Consistency

Assuming that the minimization in Equation (7) actually yields perfect alignment

$$\hat{\mathbf{y}}(\hat{\mathbf{u}}_k) = \mathbf{y}(\mathbf{u}_k),$$

the surrogate exactly satisfies 0-order consistency, i.e. $\mathbf{s}_k(\mathbf{u}_k) = \mathbf{y}(\mathbf{u}_k)$ (cf. Section 4).

If this is not the case, i.e. the minimization (7) yields a local minimum for which we would have obtained an approximate alignment only, i.e.

$$\hat{\mathbf{y}}(\hat{\mathbf{u}}_k) \approx \mathbf{y}(\mathbf{u}_k)$$

then obviously the surrogate's consistency is only satisfied approximately, i.e. $\mathbf{s}_k(\mathbf{u}_k) \approx \mathbf{y}(\mathbf{u}_k)$.

The 0-order consistency is therefore dependent on how close the alignment of the low-fidelity model can be achieved by $\mathbf{p}$. However, using the definition of the surrogate and the mapping from Equation (7), the surrogate obviously is at least as close to the high-fidelity model as the low-fidelity model itself, i.e.

$$\|\mathbf{s}_k(\mathbf{u}_k) - \mathbf{y}(\mathbf{u}_k)\| \;=\; \|\hat{\mathbf{y}}\left[\mathbf{p}(\mathbf{u}_k)\right] - \mathbf{y}(\mathbf{u}_k)\| \;\leq\; \|\hat{\mathbf{y}}(\mathbf{u}_k) - \mathbf{y}(\mathbf{u}_k)\|, \tag{8}$$

where the second relation is ensured by the minimization (7). Figure 3 illustrates this property showing the high- and low-fidelity model output as well as the surrogate model output for the state detritus at a randomly chosen parameter vector $\mathbf{u}_k$. This supports the argumentation above: In the point $\mathbf{u}_k$ the surrogate obviously provides a reasonable approximation for the high-fidelity model while being closer to it than the low-fidelity model itself. We will see in the next section that this property is also given in a neighborhood.

7 Aggressive Space Mapping

The Aggressive Space Mapping (ASM) algorithm we use in this paper was developed by Bandler et al. [1, 2]. It firstly solves for the coarse model optimum

$$\hat{\mathbf{u}}^* := \operatorname{argmin}_{\mathbf{u} \in U} J\left(\hat{\mathbf{y}}(\mathbf{u}), \mathbf{u}\right) \tag{9}$$

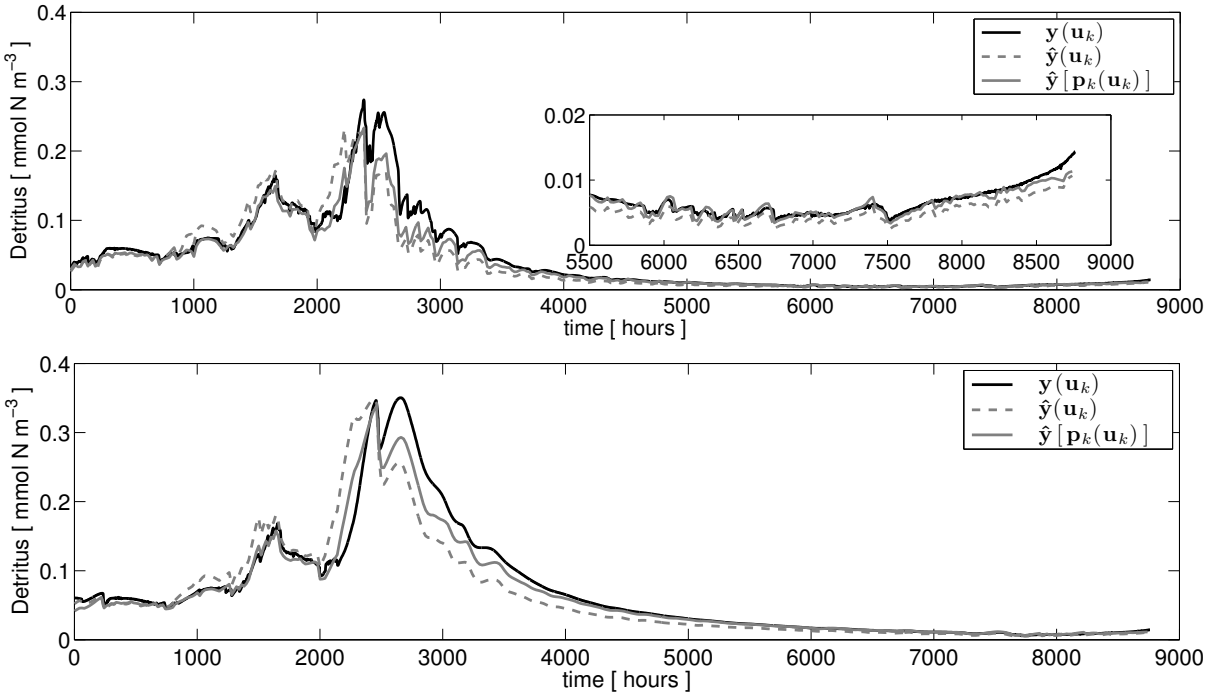

Figure 3: High- and low-fidelity model output $\mathbf{y}^{(D)}, \hat{\mathbf{y}}^{(D)}$ as well as the aligned surrogate $\mathbf{s}_k^{(D)}(\mathbf{u}_k) = \hat{\mathbf{y}}^{(D)}\,[\mathbf{p}_k(\mathbf{u}_k)]$ for the state detritus, at the same randomly chosen parameter vector $\mathbf{u}_k$, at depths $z \approx 25m$ (top) and $z \approx 60m$. The surrogate model provides a reasonable approximation of the high-fidelity model while lying closer than the low-fidelity model itself.

and then iteratively computes a solution $\bar{\mathbf{u}}$ of the nonlinear system

$$\mathbf{F}(\bar{\mathbf{u}}) \;:=\; \mathbf{p}(\bar{\mathbf{u}}) - \hat{\mathbf{u}}^* \;=\; 0. \tag{10}$$

For this purpose, a Quasi-Newton iteration [6, 7] with a Broyden rank-one approximation [3] for the Jacobian $B_k \approx \mathbf{p}'(\mathbf{u}_k)$ is used, i.e. the following steps are performed for $k = 0, 1, \ldots$:

$$\begin{aligned} \text{Solve} \quad B_k\,\mathbf{d}_k &= -\mathbf{F}(\mathbf{u}_k) = -\left(\mathbf{p}(\mathbf{u}_k) - \hat{\mathbf{u}}^*\right) \\ \mathbf{u}_{k+1} &= \mathbf{u}_k + \mathbf{d}_k \\ B_{k+1} &= B_k + \frac{(\mathbf{y}_k - B_k\,\mathbf{d}_k)\;\mathbf{d}_k^\top}{\mathbf{d}_k^\top\,\mathbf{d}_k}. \end{aligned} \tag{11}$$

The idea behind this approach (as was illustrated e.g. in [4]) is the following: If either the fine model nearly matches the data in an optimum $\mathbf{u}^*$ (i.e. $\mathbf{y}(\mathbf{u}^*) \approx \mathbf{y}_d$) or both models are similar near their respective optima ($\hat{\mathbf{y}}(\hat{\mathbf{u}}^*) \approx \mathbf{y}(\mathbf{u}^*)$), we have

$$\hat{\mathbf{y}}\,[\mathbf{p}(\mathbf{u}^*)] \approx \mathbf{y}(\mathbf{u}^*) \approx \mathbf{y}_d.$$

Then we expect the mapping to satisfy

$$\mathbf{p}(\mathbf{u}^*) \;=\; \text{argmin}_{\mathbf{u}\in U}\;||\,\hat{\mathbf{y}}(\mathbf{u}) - \mathbf{y}(\mathbf{u}^*)\,||_Y^2 \;\approx\; \text{argmin}_{\mathbf{u}\in U}\;||\,\hat{\mathbf{y}}(\mathbf{u}) - \mathbf{y}_d\,||_Y^2 = \hat{\mathbf{u}}^*, \tag{12}$$

which is also referred to as a *perfect mapping*. Solving the nonlinear system (10) is also referred to as the *original* SM approach (cf. [4]).

It was also shown in [4] that, if the mapping is injective and the coarse model optimum $\hat{\mathbf{u}}^*$ is unique, then the solution of the ASM approach coincides with the solution $\bar{\mathbf{u}}_s$ obtained by directly optimizing the surrogate defined in Equation (7), i.e.

$$\bar{\mathbf{u}}_s \;=\; \text{argmin}_{\,\mathbf{u}\in U}\; J\left(\hat{\mathbf{y}}\left[\mathbf{p}(\mathbf{u})\right], \mathbf{u}\right). \tag{13}$$

In general one cannot expect the ASM solution $\bar{\mathbf{u}}$ and $\bar{\mathbf{u}}_s$ from (13) to coincide with the fine model optimum $\mathbf{u}^*$ unless the mapping is perfect and injective. However, in general one expects fine and coarse model to show important similarities and hence to satisfy consistency at their respective optima, i.e. $\mathbf{y}(\mathbf{u}^*) \approx \hat{\mathbf{y}}(\hat{\mathbf{u}}^*)$, leading to an almost perfect mapping $\mathbf{p}(\mathbf{u}^*) \approx \hat{\mathbf{u}}^*$ and hence $\bar{\mathbf{u}} \approx \bar{\mathbf{u}}_s \approx \mathbf{u}^*$. For a more detailed analysis we refer to [4].

Globalized Quasi-Newton Method

Since the standard Quasi-Newton Algorithm, as given in the iteration (11), may suffer from local convergence one can additional use a classical line search strategy introducing a merit function $h : U \to \mathbb{R}$ given as

$$h(\mathbf{u}) \;:=\; \frac{1}{2}\,||\,\mathbf{F}(\mathbf{u})\,||^2 \;=\; \frac{1}{2}\,||\,\mathbf{p}(\mathbf{u}) - \hat{\mathbf{u}}^*\,||^2.$$

We then have

$$\nabla h(\mathbf{u}) = \mathbf{F}'(\mathbf{u})^\top \mathbf{F}(\mathbf{u})$$

(with the Jacobian $\mathbf{F}'(\mathbf{u})$). If $\mathbf{F}'(\mathbf{u}_k)B_k^{-1}$ is positive-definite, then

$$\nabla h(\mathbf{u}_k)^\top \mathbf{d}_k \;=\; \mathbf{F}(\mathbf{u}_k)^\top \mathbf{F}'(\mathbf{u}_k) B_k^{-1} \mathbf{F}(\mathbf{u}_k) \;\leq\; 0,$$

i.e. d_k is a descent direction for h at the point $\mathbf{u}_k$.

Obviously the Newton direction (where B_k is replaced by $\mathbf{F}'(\mathbf{u}_k)$) is always a descent direction for h in $\mathbf{u}_k$, satisfying $\nabla h(\mathbf{u}_k)^\top \mathbf{d}_k = -2h(\mathbf{u}_k)$. Assuming that B_k is a "good" approximation of $\mathbf{F}'(\mathbf{u_k})$, we use the last relation also in a line search in the Quasi-Newton method. The iteration step then takes the following form:

$$\begin{aligned} \text{Find } \sigma \in]0,1[\quad \text{s.t.} \quad h(\mathbf{u}_k + \sigma \mathbf{d}_k) &\leq (1 - 2\sigma\delta) h(\mathbf{u}_k) \\ &\approx h(\mathbf{u}_k) + \sigma\delta \left(\nabla h(\mathbf{u}_k)^\top \mathbf{d}_k\right) \\ \text{Update} \quad \mathbf{u}_{k+1} &= \mathbf{u}_k + \sigma \mathbf{d}_k \\ B_{k+1} &= B_k + \frac{(\mathbf{y}_k - \sigma B_k \mathbf{d}_k)\,\sigma \mathbf{d}_k^\top}{(\sigma \mathbf{d}_k^\top)(\sigma \mathbf{d}_k)}. \end{aligned} \tag{14}$$

Here $\delta \in]0,1[$ is a parameter that defines the rate of decrease in the merit function that is desired in the current step, similar as in Armijo's rule, cf. for example [7]. Since we are using an approximation in (14) anyway, we may write it in a simpler form with $2\sigma\delta$ replaced by $C \in]0,1[$ (cf. [6]). The resulting pseudo code can be found in Algorithm 1 at the end of this section.

In general, the Broyden update does *not* guarantee to provide a descent direction, and thus the line search might fail (cf. line 5,6 in Algorithm 1). In this case one could use the Newton or steepest descent direction (cf. lines 7-17 in Algorithm 1) and apply one of them directly to the merit function, following the idea that minimizing h will lead closer to a zero of $\mathbf{F}$.

Practical Issues

For a given optimization problem one has to carefully consider how many iterations and hence evaluations of the function $\mathbf{F}$ are affordable. The alignment of the low-fidelity model through the parameter mapping $\mathbf{p}$ and hence the evaluation of the function $\mathbf{F}$ is quite expensive due to the minimization required to obtain $\mathbf{p}(\mathbf{u}_k)$. One evaluation of the function $\mathbf{p}$ requires one expensive evaluation of the high-fidelity model plus the minimization which – depending on the chosen method – might additionally include derivatives or their approximations by more function evaluations of the low-fidelity model.

In order to keep the number of fine model evaluations as low as possible we did not use optional and expensive Newton or steepest descent steps but only the cheaper Quasi-Newton direction. For the results provided in the next section we furthermore used a line search with three iterations at maximum (corresponding to a minimal step length of $|| \mathbf{d} || = 1.25e - 1$). If the line search failed to find a suitable step length, the algorithm was terminated. In this case it either got stuck in a local minimum of h or the Broyden matrix provided an inaccurate approximation resulting in a non-descent direction.

Note that a successful line search in the ASM algorithm also indirectly confirms that the Taylor approximation we use to approximate the mapping $\mathbf{p}$ (cf. Eq. (7)) is reasonable in the current point $\mathbf{u}_k$ and also in some neighborhood $\mathbf{u}_k + \epsilon$, i.e.

$$\mathbf{p}_k(\mathbf{u}_k + \epsilon) \approx \mathbf{p}(\mathbf{u}_k + \epsilon)$$

for sufficiently small perturbation vectors ϵ. Hence the surrogate $\mathbf{s}_k = \hat{\mathbf{y}}\left[\mathbf{p}_k(\mathbf{u})\right]$ will be a reasonable approximation also in a neighborhood around $\mathbf{u}_k$ (cf. Eq. (8)), i.e.

$$|| \hat{\mathbf{y}}\left[\mathbf{p}_k(\mathbf{u}_k + \epsilon)\right] - \mathbf{y}(\mathbf{u}_k + \epsilon) ||^2 \quad \leq \quad || \hat{\mathbf{y}}(\mathbf{u}_k + \epsilon) - \mathbf{y}(\mathbf{u}_k + \epsilon) ||^2$$

This is also interesting when directly optimizing the surrogate (cf. Eq. (13) and end of Section (7)).

Algorithm 1 Globalized Quasi-Newton algorithm to solve for the root of $\mathbf{F}(\mathbf{u})$.

function $[\,\mathbf{u}_k\,]$ = GLOB_QUASI_NEWTON ▷ Main program

  $B_0 = I \quad , \quad \mathbf{u}_0 = \hat{\mathbf{u}}^* \quad , \quad k = 0 \quad , \quad \epsilon_h = 1e-8 \quad , \quad \epsilon_{LS} = 1.25e-1 \quad , \quad \delta = 1e-4 \quad , \quad k_{max} = 10$

  $\mathbf{F}_0 = \mathbf{F}(\mathbf{u}_0) \quad , \quad h_0 = h(\mathbf{u}_0)$ ▷ Initialization

  while $h_k \geq \epsilon_h$ **and** $k < k_{max}$ **do**

    $B_k\,\mathbf{d}_k = -\mathbf{F}_k \quad , \quad p_k = 2 \cdot \delta \cdot h_k$ (**or** $p_k = C$) ▷ Calculate Quasi-Newton step

    $[\,\mathbf{F}_{trial}, h_{trial}, \mathbf{u}_{trial}, flag\,]$ = line search$(\mathbf{u}_k, \mathbf{d}_k, h_k, p_k)$

    if $flag = 0$ **then** ▷ If line search failed ...

      Error('Quasi-Newton step failed: Step length σ below given threshold ϵ_{LS}.
        Broyden matrix might be inaccurate, i.e. $B_k \neq \mathbf{F}'_k$ or local minimum of h reached.')

      $B_k = \mathbf{F}'_k$ ▷ Approximate Jacobian

      if $\mathbf{F}'_k$ regular **then**

        $B_k\,\mathbf{d}_k = -\mathbf{F}_k \quad , \quad p_k = 2 \cdot \delta \cdot h_k$ ▷ Newton step

      else

        $\mathbf{d}_k = -B_k^\top\,\mathbf{F}_k \quad , \quad p_k = \delta \cdot ||\,\mathbf{d}_k\,||^2$ ▷ Steepest Descent step

      end if

      $[\,\mathbf{F}_{trial}, h_{trial}, \mathbf{u}_{trial}, flag\,]$ = line search$(\mathbf{u}_k, \mathbf{d}_k, h_k, p_k)$

    end if

    if $flag = 1$ **then** ▷ Step successful

      $\mathbf{F}_{k+1} = \mathbf{F}_{trial} \quad , \quad h_{k+1} = h_{trial} \quad , \quad \mathbf{u}_{k+1} = \mathbf{u}_{trial} \quad , \quad \mathbf{d}_k = \mathbf{u}_{k+1} - \mathbf{u}_k$

      $B_{k+1} = B_k + \frac{(\mathbf{y}_k - B_k\,\mathbf{d}_k)\,\mathbf{d}_k^\top}{\mathbf{d}_k^\top\,\mathbf{d}_k}$

      $k = k + 1$

    else

      Error('Also Newton/ Steepest Descent step failed: Step length below given threshold ϵ_{LS}.
        Local minimum assumed.')

      $k = k_{max}$

    end if

  end while

end function

function $[\,\mathbf{F}_{trial}, h_{trial}, \mathbf{u}_{trial}, flag\,]$ = LINE SEARCH$(\mathbf{u}, \mathbf{d}, \mathbf{h}, p_k)$ ▷ line search procedure

  $\alpha = 0 \quad , \quad \sigma = \frac{1}{2}^{\alpha} \quad , \quad flag = 0$ ▷ Initialization

  while $(\,\sigma > \epsilon_{LS}$ **and** $flag = 0\,)$ **do**

    $\mathbf{u}_{trial} = \mathbf{u} + \sigma \cdot \mathbf{d} \quad , \quad \mathbf{F}_{trial} = \mathbf{F}(\mathbf{u}_{trial}) \quad , \quad h_{trial} = h(\mathbf{u}_{trial})$

    if $h_{trial} \leq h - \sigma \cdot p_k$ **then**

      $flag = 1$ ▷ $flag$ indicates success/ failure of line search

    else

      $\alpha = \alpha + 1 \quad , \quad \sigma = \frac{1}{2}^{\alpha}$

    end if

  end while

end function

Definitions:

$\mathbf{F}(\mathbf{u}_k) := \mathbf{p}(\mathbf{u}_k) - \hat{\mathbf{u}}^* \quad , \quad h(\mathbf{u}_k) := 1/2 \cdot ||\,\mathbf{F}(\mathbf{u}_k)\,||^2$

$\mathbf{p}(\mathbf{u}_k) = \min_{\mathbf{u} \in U} ||\,\hat{\mathbf{y}}(\mathbf{u}) - \mathbf{y}(\mathbf{u}_k)\,||_Y^2 \quad , \quad \mathbf{F}_k := \mathbf{F}(\mathbf{u}_k) \quad , \quad h_k := h(\mathbf{u}_k) \quad , \quad \mathbf{y}_k := \mathbf{F}_{k+1} - \mathbf{F}_k$

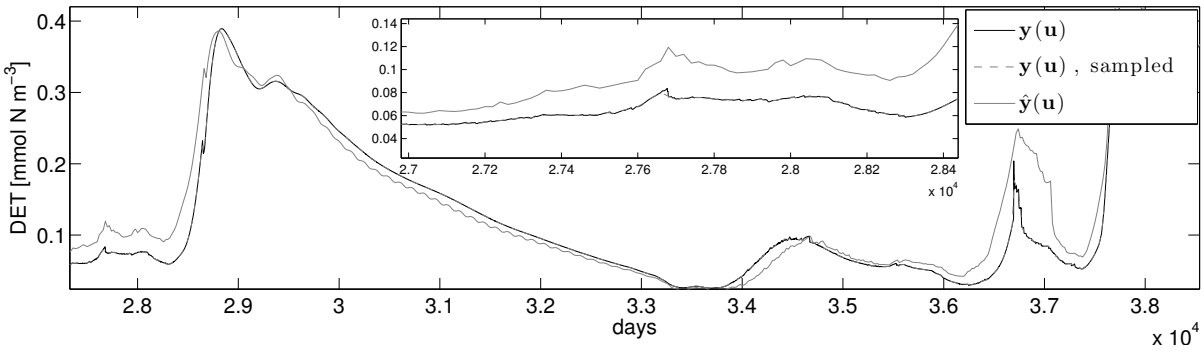

Figure 4: Comparison of the full (i.e. not sampled) fine model output $y^{(D)}(z_i, t_j, \mathbf{u})$ for the state detritus, the corresponding sampled output $y^{(D)}(z_i, t_{\beta j}, \mathbf{u})$ and the coarse model output $\hat{y}^{(D)}(z_i, t_j, \mathbf{u})$ at the same randomly chosen parameter vector $\mathbf{u}$, in the 10th vertical layer, using a coarsening factor of $\beta = 20$. Curves for original and sampled fine model output are very close.

8 Results and Discussion

We tested the ASM approach by using synthetic target data $\mathbf{y}_d = \mathbf{y}(\mathbf{u}^{opt})$ at a randomly chosen parameter vector $\mathbf{u}^{opt}$ within admissible bounds $\mathbf{b}_u, \mathbf{b}_l$. We used the globalized Quasi-Newton method described in the last section. For the optimization of the low- and high-fidelity model and for the minimization required to obtain the parameter mapping $\mathbf{p}$ (7) we used the MATLAB* function `fmincon`, taking the option for the active-set algorithm. The cost function (using a weighted Euclidean vector norm) is given as follows (here considering the high-fidelity model output $\mathbf{y}$)

$$J\left(\mathbf{y}(\mathbf{u}), \mathbf{u}\right) = \frac{1}{N_{tracers} K M} \sum_{l=N,P,Z,D} \sum_{i=1}^{K} \sum_{j=1}^{M} \left(y^{(l)}(z_i, t_j, \mathbf{u}) - y_d^{(l)}(z_i, t_j) \right)^2, \tag{15}$$

where $N_{tracers} = 4$ denotes the number of tracers.

Data Sampling

For the low-fidelity model we accordingly use a sum over $\hat{M}$ discrete time steps (cf. Section 5) in the cost function J and take into account only every βth time step of the target data $\mathbf{y}_d$, hence replacing $y_d^{(l)}(z_i, t_j)$ by the sampled output $y_d^{(l)}(z_i, t_{\beta j})$. For obtaining the parameter mapping $\mathbf{p}$ we analogously sum over the discrete time steps of the low-fidelity model output $\hat{\mathbf{y}}$, hence considering in this case the sampled high-fidelity model output $\mathbf{y}$:

$$\mathbf{p}(\mathbf{u}_k) = \text{argmin}_{\mathbf{u} \in U} \left[\frac{1}{N_{tracers} K \hat{M}} \sum_{l=N,P,Z,D} \sum_{i=1}^{K} \sum_{j=1}^{\hat{M}} \left(\hat{y}^{(l)}(z_i, t_j, \mathbf{u}) - y^{(l)}(z_i, t_{\beta j}, \mathbf{u}_k) \right)^2 \right].$$

Figure 4 shows a comparison of the full (i.e. not sampled) fine model output $y(z_i, t_j, \mathbf{u})$, the corresponding sampled output $y(z_i, t_{\beta j}, \mathbf{u})$ and the coarse model output $\hat{y}(z_i, t_j, \mathbf{u})$ using a coarsening factor of $\beta = 20$ (as in the following results).

*MATLAB is a registered trademark of The MathWorks, Inc., http://www.mathworks.com

ASM vs. Direct Fine Model Optimization

We now compare the results and the computational cost of the ASM approach with the direct optimization of the high-fidelity model using the cost function given in Equation (15). To give a profound illustration of the behavior of the algorithm, we below consider the following parameter values, model outputs, and respective cost function values:

(i) The target $\mathbf{y}_d$, i.e. the fine model output at a randomly chosen parameter vector $\mathbf{u}^{opt}$,

(ii) the fine model output at another randomly chosen parameter vector $\mathbf{u}_0$, serving as initial value of the optimization runs,

(iii) the model output at the result $\hat{\mathbf{u}}^*$ of a pure coarse model optimization,

(iv) the fine model output at the result $\bar{\mathbf{u}}$ of the ASM algorithm, and

(v) the output of a (rather expensive) pure fine model optimization yielding $\mathbf{u}^*$.

Using different optimization routines might yield different results in (iii-v), but this will probably not influence the relative reduction in the total optimization cost using the ASM algorithm (see below). For example, in [9] better cost function values were obtained by direct fine model optimization using a different optimization method (not MATLAB's `fmincon`) for the same problem and the same model.

A good agreement between the results of (i) and (v) would indicate a high quality of the used optimization method itself, whereas a good agreement between those from (iv) and (v) would mean that the ASM works fine.

We experienced that results for different initial parameter vectors $\mathbf{u}_0$ are comparable. For illustration we here present the results of two exemplary test runs, considering the same target data set $\mathbf{y}_d$ (i), but different initial parameters $\mathbf{u}_0$ (ii). In both cases we used $K = 20, M = 8760 \cdot 5$ and $\beta = 20$. Hence for the low-fidelity model we obtain $\hat{M} = M/\beta = 2190$ discrete time steps per spatial grid point z_i.

In Figures 5 and 6 below we only show some tracers for a part of the whole time interval (one year) at some distinct depth layers. Moreover the output of the fine model was sampled as described in the last subsection. The qualitative behavior of the other tracers and at different times and spatial layers is similar. Table 1 shows the resulting values of the parameters, the cost function and the reduction in computational time.

Test Results for Model Output and Computational Cost

Figure 5 illustrates the results of the ASM and the high and low-fidelity model for a first choice of the initial guess $\mathbf{u}_0$. Corresponding parameters and values of the cost function J are given in the upper part of Table 1. Furthermore the table shows the total cost of the high-fidelity ($C_{opt,h}$) and the low-fidelity optimization ($C_{opt,l}$) and of the Quasi-Newton iterations of the ASM algorithm (C_{QN}) in terms of the total number of equivalent fine model evaluations, which were required to reach the given value of the cost function J. Note that the total cost in the ASM approach consists of the cost for the low-fidelity model optimization $C_{opt,l}$ and those for solving the nonlinear system of equations by the Quasi-Newton method, i.e. C_{QN}. For details see also the next subsection.

From Figure 5 we see that by direct fine model optimization we yield a very reasonable optimal fit $\mathbf{y}(\mathbf{u}^*)$ (grey dashed line) of the target data $\mathbf{y}_d$ (black line). This is equivalent to

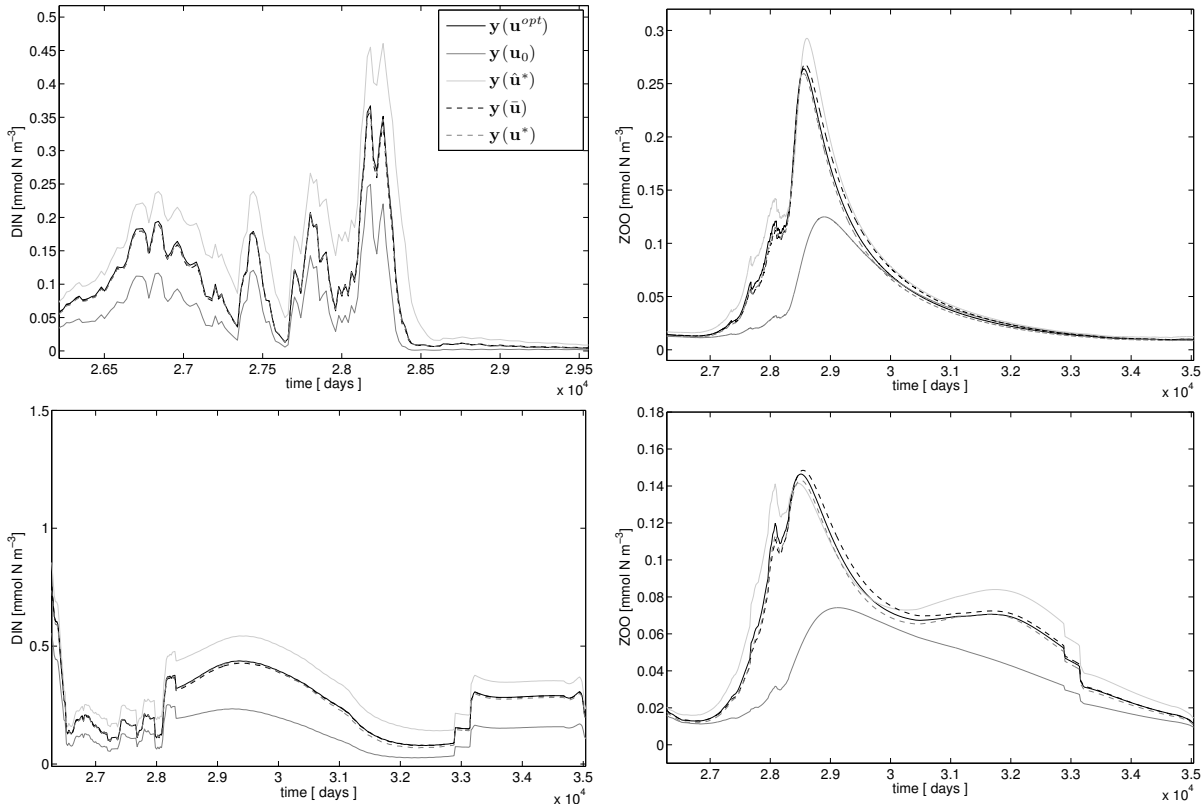

Figure 5: Model output $\mathbf{y}^{(N)}$ (detritus, left) and $\mathbf{y}^{(Z)}$ (zooplankton, right) at depth $z \approx 2.68m$ (top) and $z \approx 108.15m$ (bottom). Shown are, in the legend from top to bottom: (i) Target $\mathbf{y}_d$, i.e. fine model output at randomly chosen parameters $\mathbf{u}^{opt}$, (ii) fine model output at the initial value $\mathbf{u}_0$, (iii) coarse model output at its optimum $\hat{\mathbf{u}}^*$, (iv) result of the ASM algorithm at its optimum $\bar{\mathbf{u}}$, and (v) result of a direct fine model optimization yielding $\mathbf{u}^*$. On the top left, we only show the interesting time interval. Curves corresponding to (i), (iv) and (v) are very close.

a cost function value of $J(\mathbf{y}(\mathbf{u}^*), \mathbf{u}^*) = 1.611e-05$ obtained after 281 function evaluations (cf. Table 1). We furthermore see that by coarse model optimization we yield parameters $\hat{\mathbf{u}}^*$ with a fit $\mathbf{y}(\hat{\mathbf{u}}^*)$ (light grey line) which obviously provides only a rough approximation of the target data, but in $C_{opt,l} = 19.95$ equivalent fine model evaluations only. Using the ASM approach, we finally obtain a solution $\bar{\mathbf{u}}$ with an optimal fit $\mathbf{y}(\hat{\mathbf{u}}^*)$ (black dashed line) and parameter match lying very close to that obtained by the fine model optimization.

The key point now is that the ASM solution $\bar{\mathbf{u}}$ is obtained in only $C_{QN} = 80.25$ equivalent fine model evaluations for the Quasi-Newton steps. Summarizing, using the ASM approach, we hence obtained a very reasonable solution in totally

$$C_{ASM} \quad := \quad C_{opt,l} + C_{QN} \approx 100$$

equivalent fine model evaluations. The same cost function value by direct fine model optimization was obtained after 236 model evaluations (cf. Table 1). This leads to a reduction in the total optimization cost of about 57%.

Figure 6 and the lower part of Table 1 show the corresponding results of another run with different initial parameters $\mathbf{u}_0$. Obviously results of the fine and coarse model optimization look similar. Also a similar convenient fit was obtained by the ASM approach. Here we required about 117 equivalent fine model evaluations to obtain the ASM solution while here about 409 evaluations were necessary to yield the same cost function value by

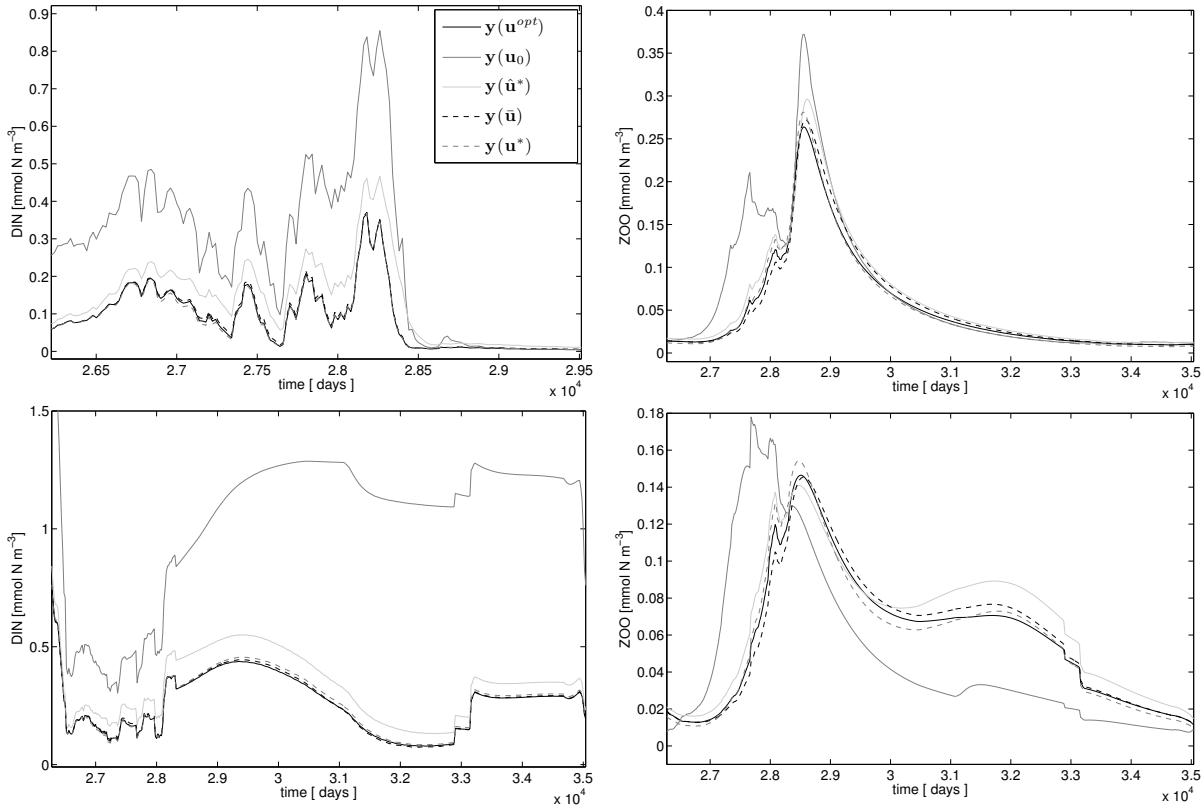

Figure 6: Same as Figure 5, but for a different initial guess $\mathbf{u}_0$ for the optimization runs. Again the curves corresponding to (i), (iv) and (v) described on page 8 are very close.

fine model optimization. This leads to an even better reduction in the total optimization cost of about 71%.

Analysis of the Optimization Cost

In order to compare the total optimization cost C_{ASM} of the ASM approach with the one obtained from the fine model optimization ($C_{opt,h}$) we consider in both cases the cost in terms of total number of equivalent fine model evaluations. We generally yield the following

$$\begin{aligned} C_{ASM} &= C_{opt,l} + C_{QN} \quad , \quad C_{QN} = N_{ASM} \cdot C_p \cdot N_{LS}^{qn} \quad , \\ C_p &= C_{align} + 1 = N_{opt,p} \cdot (C_{grad} + N_{LS}^{opt})/\beta + 1 \quad , \\ C_{opt,l} &= N_{func,l} = N_{opt,l} \cdot (C_{grad} + N_{LS}^{opt})/\beta \quad , \\ C_{opt,h} &= N_{func,h} = N_{opt,h} \cdot (C_{grad} + N_{LS}^{opt}) \quad , \quad C_{grad} = 12 \quad . \end{aligned} \tag{16}$$

Here $N_{func,h}, N_{func,l}$ denote the number of fine model evaluations needed in the optimization procedures optimizing the fine respectively the coarse model. This is further given as the number of iterations, denoted by $N_{opt,h}, N_{opt,l}$, times the cost of the gradient C_{grad} plus the number of line search steps done per iteration, denoted by N_{LS}^{opt}. For the cost for a gradient evaluation we use the number of optimization variables (i.e. 12 here), which corresponds to the usual effort for a finite difference approximation and that for a forward mode AD (Algorithmic/Automatic Differentiation) gradient.

iterate	$u_{k,1}$ $u_{k,2}$ ... $u_{k,12}$	$J(\mathbf{y}(\mathbf{u}),\mathbf{u})$	C_i
$\mathbf{u}_0$	0.4862 0.6442 0.0192 0.0101 0.0372 0.9333 1.9045 0.0063 0.1804 0.0170 0.4060 6.9374	5.885e-03	
	Fine model optimization: $\mathbf{u}^* := \text{argmin}_{\mathbf{u}\in U} J(\mathbf{y}(\mathbf{u}),\mathbf{u})$		
$\mathbf{u}^*$	0.7635 0.5989 0.0274 0.0102 0.0352 1.0182 1.9300 0.0104 0.2179 0.0196 0.4953 5.8663	1.611e-05	281
	Coarse model optimization: $\hat{\mathbf{u}}^* := \text{argmin}_{\mathbf{u}\in U} J(\hat{\mathbf{y}}(\mathbf{u}),\mathbf{u})$		
$\hat{\mathbf{u}}^*$	0.7594 0.3631 0.0254 0.0119 0.0291 1.1181 0.8639 0.0070 0.1941 0.0158 0.4905 5.4200	1.751e-03	19.95
	ASM: Solve $\mathbf{F}(\bar{\mathbf{u}}) := \mathbf{p}(\bar{\mathbf{u}}) - \hat{\mathbf{u}}^* = 0$		
$\bar{\mathbf{u}}$	0.7587 0.5865 0.0265 0.0109 0.0340 0.9440 1.5241 0.0102 0.1790 0.0198 0.4895 6.0725	5.003e-05	80.25
$\mathbf{u}^{opt}$	0.7500 0.6000 0.0250 0.0100 0.0300 1.0000 2.0000 0.0100 0.2050 0.0200 0.5000 6.0000	57.54% reduction	

iterate	$u_{k,1}$ $u_{k,2}$ ... $u_{k,12}$	$J(\mathbf{y}(\mathbf{u}),\mathbf{u})$	C_i
$\mathbf{u}_0$	0.5648 0.6720 0.0153 0.0117 0.0362 1.0959 2.3349 0.0126 0.2089 0.0276 0.4524 5.2346	7.027e-02	
	Fine model optimization: $\mathbf{u}^* := \text{argmin}_{\mathbf{u}\in U} J(\mathbf{y}(\mathbf{u}),\mathbf{u})$		
$\mathbf{u}^*$	0.8709 0.5933 0.0291 0.0121 0.0382 1.0478 0.9517 0.0112 0.2233 0.0192 0.4663 5.8359	5.600e-05	418
	Coarse model optimization: $\hat{\mathbf{u}}^* := \text{argmin}_{\mathbf{u}\in U} J(\hat{\mathbf{y}}(\mathbf{u}),\mathbf{u})$		
$\hat{\mathbf{u}}^*$	0.7588 0.3558 0.0294 0.0119 0.0365 1.1379 0.8479 0.0070 0.1881 0.0158 0.5023 5.4750	1.832e-03	26.35
	ASM: Solve $\mathbf{F}(\bar{\mathbf{u}}) := \mathbf{p}(\bar{\mathbf{u}}) - \hat{\mathbf{u}}^* = 0$		
$\bar{\mathbf{u}}$	0.7607 0.5715 0.0313 0.0109 0.0429 0.9598 1.5285 0.0109 0.1740 0.0195 0.5115 5.9755	5.889e-05	91.15
$\mathbf{u}^{opt}$	0.7500 0.6000 0.0250 0.0100 0.0300 1.0000 2.0000 0.0100 0.2050 0.0200 0.5000 6.0000	71.27% reduction	

Table 1: Results of the high- and low-fidelity model optimization and of the ASM algorithm from two illustrative test runs, corresponding to Figures 5 (top) and 6 (bottom), See the text for details. Also shown are the corresponding values of the cost function J and the computational cost C_i in terms of the total number of equivalent fine model evaluations required to obtain the given cost function value J, again for the three cases, i.e. $C_i \in \{C_{opt,h}, C_{opt,l}, C_{QN}\}$.

As we described in Section 7, the ASM algorithm involves firstly to solve for the coarse model optimum $\hat{\mathbf{u}}^*$ resulting in $N_{func,l}$ equivalent fine model evaluations. The second part within the ASM algorithm involves the (globalized) Quasi-Newton iteration (cf. Section 7) which results in $C_{QN} = N_{ASM} \cdot C_p \cdot N_{LS}^{qn}$ equivalent fine model evaluations. Here N_{ASM} denotes the overall number of steps in the ASM, i.e. the number of Quasi-Newton iterations, C_p denotes the cost of calculating the mapping $\mathbf{p}$ and N_{LS}^{qn} denotes the number of line search steps. The cost of the mapping, C_p, is furthermore given by one fine model evaluation plus the cost of the minimization required for the mapping, which we denote by C_{align}.

In the numerical tests we obtained the following: At average $\bar{N}_{func,h} \approx 230$, $\bar{N}_{func,l} \approx 1.4 \cdot \bar{N}_{func,h}/\beta$, $\bar{C}_{align} \approx 1.4 \cdot \bar{N}_{func,h}/\beta$, $\bar{N}_{ASM} \approx 3.125$, $\bar{N}_{LS}^{qn} \approx 1.2$. With those averages we obtain the following

$$\begin{aligned} C_{ASM} &= 1.4 \cdot N_{func,h}/\beta + 3.125 \cdot (1.4 \cdot N_{func,h}/\beta + 1) \cdot 1.2, \\ \text{i.e.} \quad C_{ASM}/C_{opt,h} &= 1.4/\beta + 3.125 \cdot (1.4/\beta + 1/N_{func,h}) \cdot 1.2 \end{aligned} \tag{17}$$

Now, with a coarsening factor of $\beta = 20$ and $\bar{N}_{func,h} \approx 230$ this leads to an average reduction of $(1 - C_{ASM}/C_{opt,h}) \approx 0.65$, hence at average about 65%.

With the calculation above we illustrated how the different parts of the ASM algorithm contribute to the total cost C_{ASM}. In principle, values for N_{ASM} and N_{LS}^{qn} are up to the user and will be problem specific. Allowing for a greater number of ASM steps N_{ASM} or number of line search steps N_{LS}^{qn} in the globalized Quasi-Newton algorithm increases the total cost C_{ASM} while not necessarily yielding a more accurate solution in the end. To improve the results in terms of reduction in the total cost one might also decrease the number $N_{func,l}$ by stopping the optimization of the coarse model after a certain number of iterations, as more iterations might not necessarily yield a significant better solution. Concluding, as we already mentioned before (cf. Section 7), the setting for the algorithm has to be carefully chosen to yield a reasonable solution at a sufficiently small number of equivalent fine model evaluations.

9 Conclusions

The optimization of models that couple ocean circulation and a marine ecosystem model can be very expensive in terms of model and gradient evaluations, especially for complex 3-dimensional models. Hence methods reducing the total optimization cost, such as surrogate-based optimization techniques, are highly desirable. In this paper we successfully applied the so-called ASM approach, firstly developed in [1], to the optimization of a 1-dimensional coupled marine ecosystem model. We used a coarser discretization in time to create a reasonable low-fidelity model the ASM approach is based on. We also showed that using the mapping definition from the ASM approach, one can obviously yield a reasonable surrogate for the fine model. We recalled that the ASM approach is a conditionally equivalent approach to use this surrogate in the optimization run, replacing the fine model. Furthermore we used a globalized Quasi-Newton iteration to obtain the ASM solution. We verified our approach by using synthetic target data comparing the results to those of the direct fine model optimization. All in all we have shown that with the ASM approach we could yield a very reasonable solution within a few number of fine model evaluations only, resulting in a significant reduction in the total optimization cost of about 65% at average.

References

[1] J.W. Bandler, R.M. Biernacki, Shao Hua Chen, P.A. Grobelny, and R.H. Hemmers. Space mapping technique for electromagnetic optimization. *IEEE Transactions on Microwave Theory and Techniques*, 42(12):2536–2544, Dec 1994.

[2] J. W. Bandler, Qingsha S. Cheng, Sameh A. Dakroury, Ahmed S. Mohamed, Student Member, Student Member, Student Member, Mohamed H. Bakr, Kaj Madsen, and Jacob Søndergaard. Space mapping: The state of the art. *IEEE Transactions on Microwave Theory and Techniques*, 52(1), 2004.

[3] C. G. Broyden. A class of methods for solving nonlinear simultaneous equations. *Mathematics of Computation*, 19(92):577–593, 1965.

[4] D. Echeverría and P.W. Hemker. Space mapping and defect correction. *Computational Methods in Applied Mathematics*, 5:107–136, 2005.

[5] W. Hackbusch. *Elliptic Differential Equations: Theory and Numerical Treatment.* Springer Series in Computational Mathematics, Springer Berlin 2010.

[6] P. Kosmol. *Methoden zur numerischen Behandlung nichtlinearer Gleichungen und Optimierungsaufgaben.* Teubner, 1993.

[7] J. Nocedal and S.J. Wright. *Numerical Optimization.* Springer, New York, 2000.

[8] A. Oschlies and V. Garcon. An eddy-permitting coupled physical-biological model of the north atlantic. 1. sensitivity to advection numerics and mixed layer physics. *Global Biogeochemical Cycles*, 13:135–160, 1999.

[9] J. Rückelt, V. Sauerland, T. Slawig, A. Srivastav, B. Ward, and C. Patvardhan. Parameter optimization and uncertainty analysis in a model of oceanic co2-uptake using a hybrid algorithm and algorithmic differentiation. *Nonlinear Analysis Real World Applications* 11(5):3993-4009, 2010.